"Innovative training approach for Supportive Living Operators"

[2016-1-EL01-KA202-023612]

CareVET's sensors toolkit (O4)

Program

Strategic Partnerships for vocational education and training

Project Number	2016-1-EL01-KA202-023612	**Acronym**	CareV.E.T.
Full Title	Innovative training approach for Supportive Living Operators		
Project URL	www.carevet.eu		

Output	Number	4	Title	CareV.E.T. Vision and coherence with European VET objectives
Date of Delivery	31-12-2017	Actual	30-04-2018	
Nature	Report			
Languages	English			
Dissemination level	CareV.E.T Consortium Partners Restricted			

Partners Involved	SenseWorks LTD, FRA-UAS			
Responsible Author	Name	Alexandros Sofios	E-mail	a.sofios@senseworks.gr
	Partner	SenseWorks		
Abstract	The toolkit contains a basic set of different sensors that simulate a real-life installation in a Supported Living Residence and helps the trainees to capitalize the role and the importance of assisted living technologies to the job curriculum they have chosen to follow.			
Description	The output describes the training toolkit that was developed to be used during the trainings offering hands on experience with these new technologies.			

Version Log			
Issue Date	**Rev. No.**	**Author / Reviewer**	**Change**
7/10/2017	0.1	Alexandros Sofios	First Version of the Toolkit
11/10/2017	0.2	Evaluation during the Training in Frankfurt	Feedback
27/03/2018	0.3	Alexandros Sofios: Sending the toolkit to Italy and Germany for installation with installation manual.	
4/4/2018	0.3	Barbara Klein Maurizio Panzironi	Evaluation of the toolkit and the installation process.
30/4/2018	1.0	Alexandros Sofios	Final draft of the report
30/07/2018	1.0	Alexandros Sofios	Added some pictures from the trainings tool place with the use of the toolkit

Contents

1 Executive Summary

Digitalization of work and life is an issue also for residencies for people with intellectual disabilities. Upcoming sensor-based products allow more autonomy of residents and changing work processes for staff. Key issue is qualification in order to utilize these new possibilities.

The sensor-based toolkit developed by SenseWorks is the core component of CareV.E.T Curriculum. The toolkit is a plug and play solution for personal activity monitoring comprising of a set of sensors and wearables that can be easily setup in any living area with Wi-Fi supporting WPA2 encryption.

The toolkit combining location information from a smart watch and stationary Bluetooth beacons, data from motion and intrusion sensors offers a solution that can deliver adaptive services with regard to the knowledge of users' lifestyle. The toolkit is easy to install, requiring only to place the sensors in the living environment and to provide the local Wi-Fi Password through a web interface or a smart watch Android app.

Upon connected to the internet a rule based monitoring platform, using real time data from sensors, allows carers - even the users themselves - to set custom and personalized rules for their safety and wellbeing. Alerts and notifications are sent when the conditions of these rules are violated.

The toolkit will be tested with staff dealing with people with intellectual disabilities in Italy; at the residency for people with intellectual disabilities at Ergastiri in Greece and in the Master course Barrier-free Systems at Frankfurt UAS in Germany.

The toolkit seems to be particular helpful for training purposes as it can run as a hands-on project in any classroom. In a further extension, the toolkit aims to support independent living for elder and people with disabilities living alone or unsupervised at home.

CareV.E.T toolkit puts emphasis on the importance of deployment in real world settings as opposed to prototype testing in pilot sites and laboratories. It aims to minimize numerous instrumentation, implementation and cost barriers, thus allowing care givers and end users to take advantage of these assistive technological developments.

2 CareV.E.T. Toolkit

The CareV.E.T. toolkit is composed of:

- Internet gateway: raspberry pi
- Sensors
- Smartwatch: Sony SmartWatch 3
- Blue-Tooth Beacons
- A rule-based monitoring platform (part of 03)

Figure 1: CareVET Toolkit

The purpose of the development of the CareV.E.T. Toolkit is the trainees to:

- Understand Internet of Things (IoT).
- Assess whether a sensor is working properly.
- Understand when and why a specific wireless network is preferred.
- Correctly install and utilize sensors.
- Discover what kind of power source each sensor is using and why.

Toolkit as a teaching aid can be used in order trainees to understand:

- How is information transmitted wirelessly?
- Experience the way information flows from sensor to cloud server.
- What are embedded devices?
- How Internet of Things (IoT) combines wireless networks and embedded devices.
- How indoors vs outdoors location works.

2.1 Toolkit's Indoors location

The indoors location component of the toolkit consists of:

- Bluetooth Low Energy (BLE) beacons advertising data
- Wearable(smartwatch) BLE scanning for data
  - records beacons' RSSI
  - sends data to server (WiFi)
- Server compares data to database and figures out indoors location of the wearable device. Location depicted online.

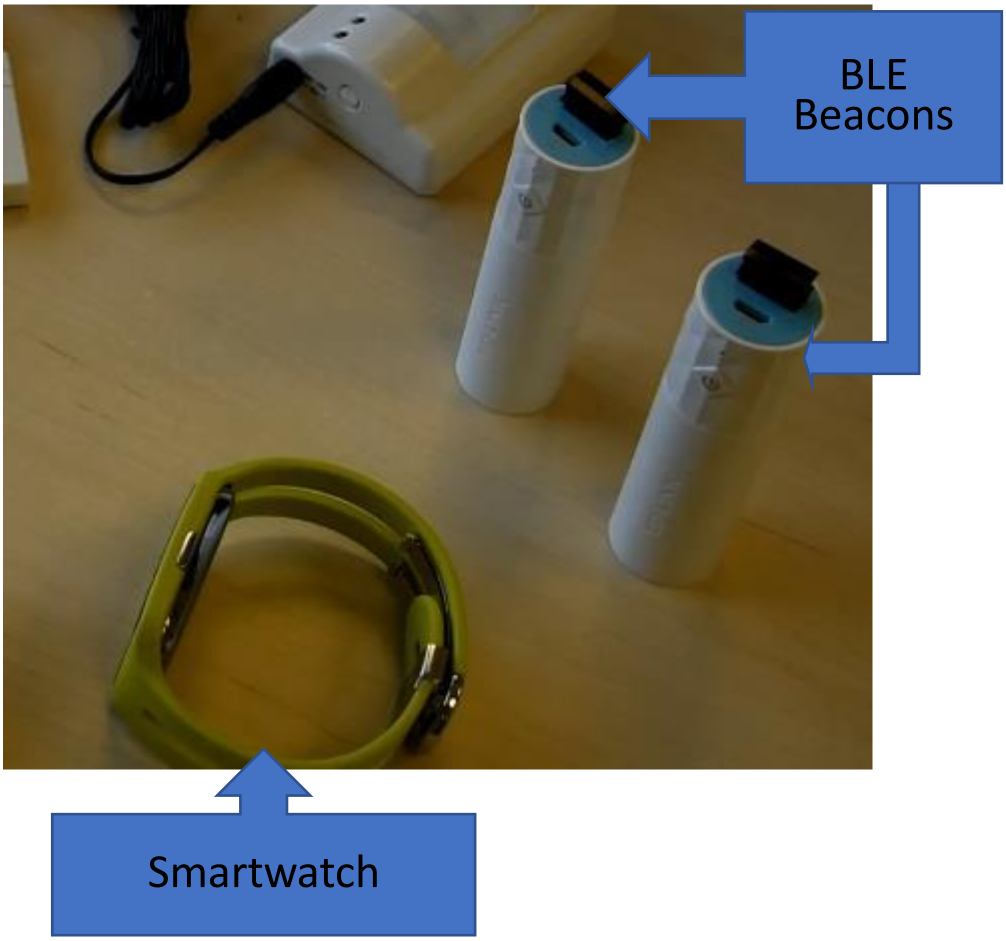


Figure 2: CareVET toolkit, indoors location component

The purpose of the component is to:

- Demonstrate how the wearable combines WiFi and BLE
- Help trainees experience the effect of noise on indoor location calculation:
  - *Signal reflection , absorption*
  - *BLE beacons' range*
  - *User's body absorption*

- Demonstrate energy consumption of WiFi vs BLE: with current setup BLE beacons can be battery powered and last for a long time

2.2 Toolkit's Sensors

The sensors component of the toolkit consists of:

- Internet gateway Raspberry Pi Zero (WiFi)
- Zigbee coordinator: dongle or Raspberry Pi
- Zigbee Motion Sensor
- Zigbee Door Sensor (magnetic reed)
- Easily expandable, not only with Zigbee sensors but with anything compatible with Raspberry Pi

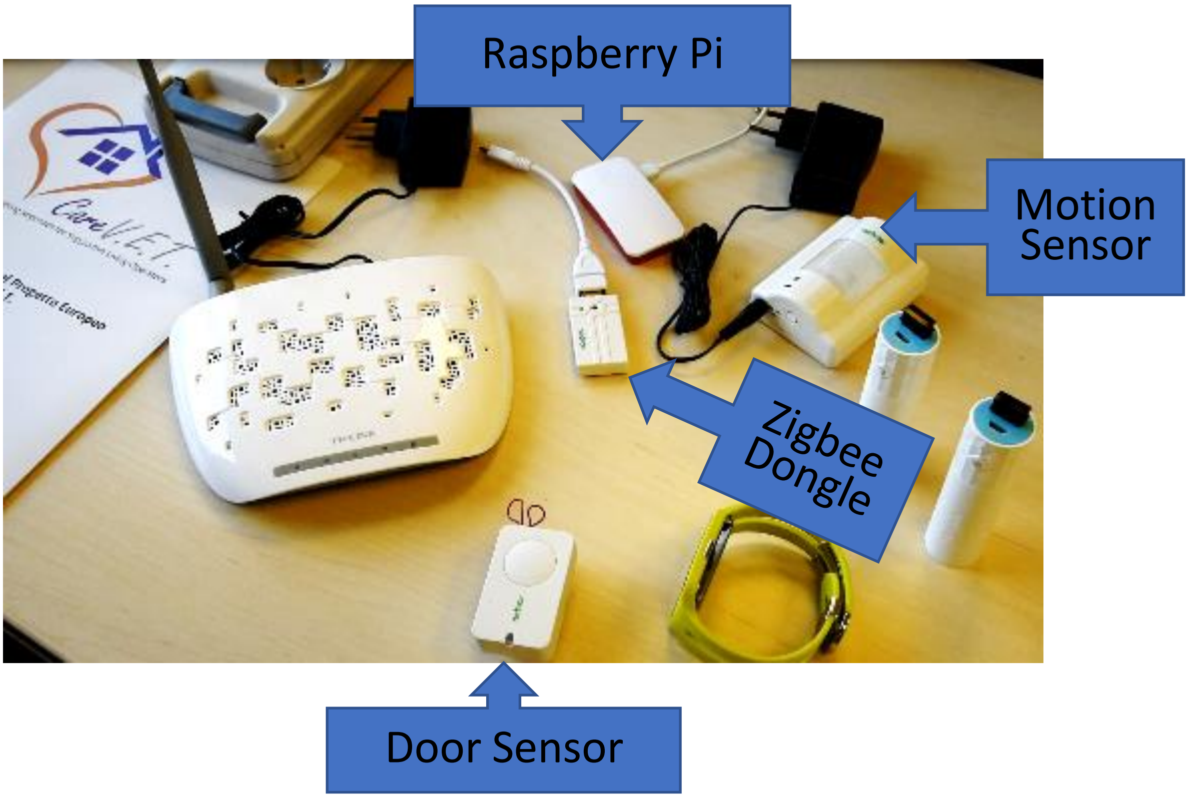


Figure 3: CareVET toolkit, sensors component

The purpose of the component is to:

- Demonstrate how multiple protocols can be bridged together: WiFi + Zigbee
- Demonstrate how end point sensors depend on a network coordinator
- Demonstrate that end point sensors are not (and should not) be always ON

Sensors are embedded systems: mini PCs that can "sleep" to save power. They can also serve as an introduction to embedded systems that make up the IoT.

2.3 Usage Example

A possible usage scenario is the Supportive Living Operator to want to monitor one's sleep.

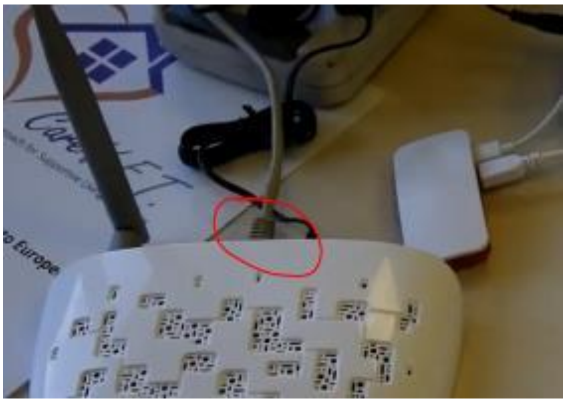
Possible solutions to that would be:

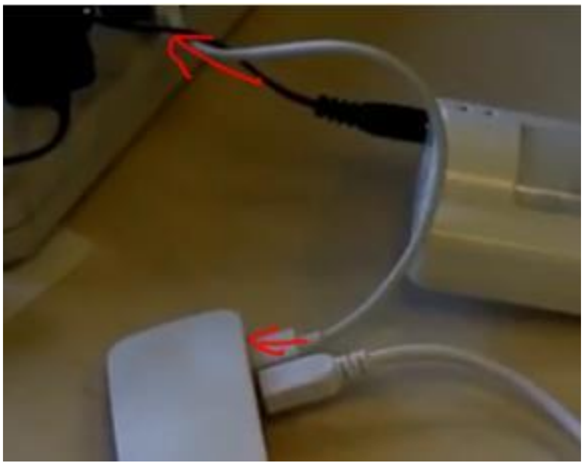
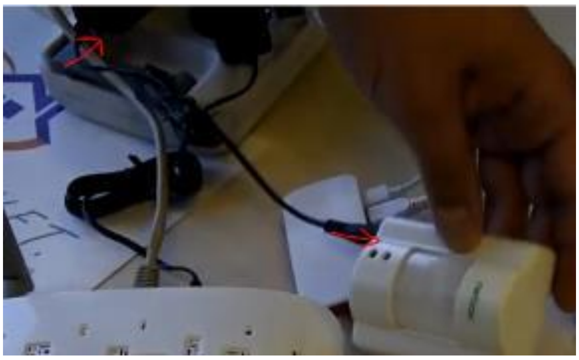
- Motion detection sensor with line of sight to one's bed
- Place a door opening sensor to one's bedroom door
- Combination of the two previous solutions above for improving accuracy and minimising false positives.
- Alerts when the user changes her/his position relevant to t the nearest beacons.

2.4 Installation

The installation of the CareVET toolkit is an easy process, also for non-experienced users. A Wifi router is additionally provided in order to facilitate further the installation process.

The process of the installation is as follows:

1. Power up the Wi-Fi Router.	
2. Connect the Wi-Fi Router to the local network through a LAN cable.	

3. Power up the Raspberry Pi Zero through the USB cable.	
4. Connect the USB dongle to the Raspberry	
5. Turn on the Door Sensor by Clicking the bottom button at the sides of the equipment. A red and then green blinking lights appear in the front of the device.	
6. Power up the motion sensor. It will automatically start blinking.	

7. Put the blue tooth beacons to the power banks to have power (check that the power banks are not empty.	
8. Open the Watch using the button at the right side (the watch must be charged)	
9. When the screen opens, drag the screen to the left, go to settings, and see if it is automatically connected to the Wi-Fi. Go to the initial screen, drag the screen to the left again and select SMARTWATCHBLE application and start it. Click location BLE to start checking for beacons nearby. When a new beacon is approached a msg appear. When you finish testing go to settings and click power off to save the Smartwatch battery when you are not using it.	

A video for the installation process is available on YouTube.

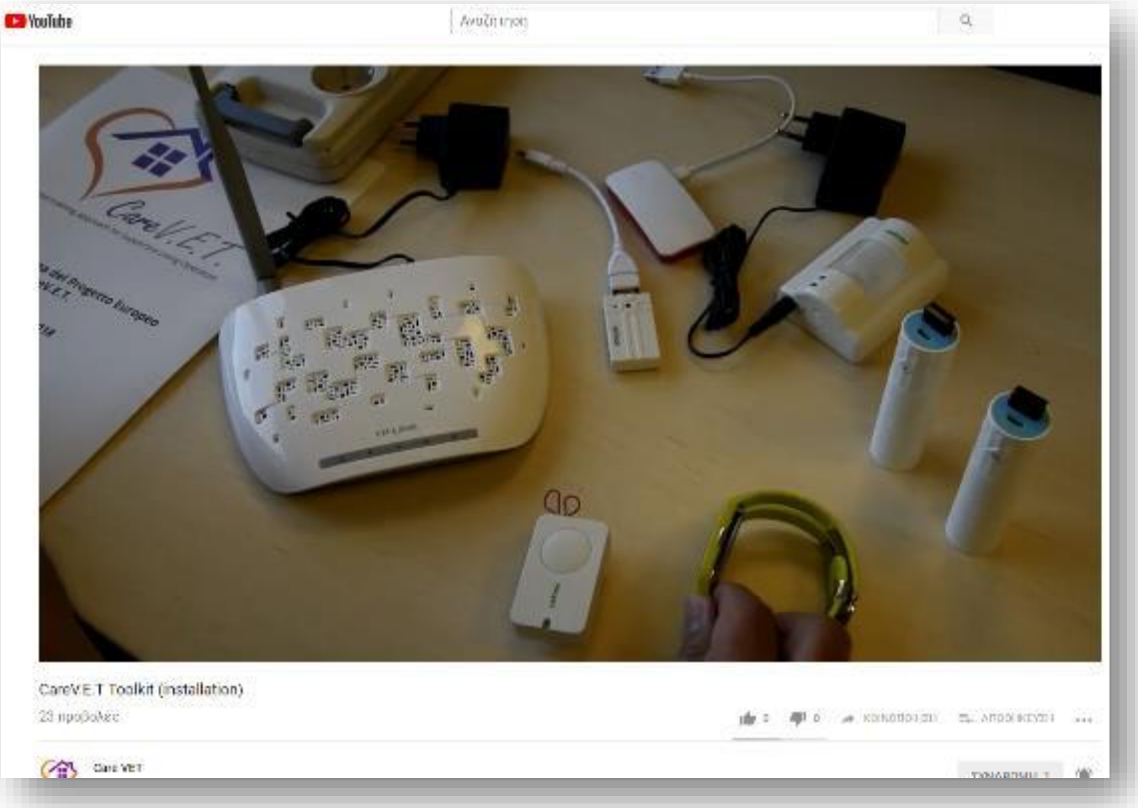


Figure 4: Video on the toolkit Installation process

The toolkit was presented live in different occasions like the conference in Rome.

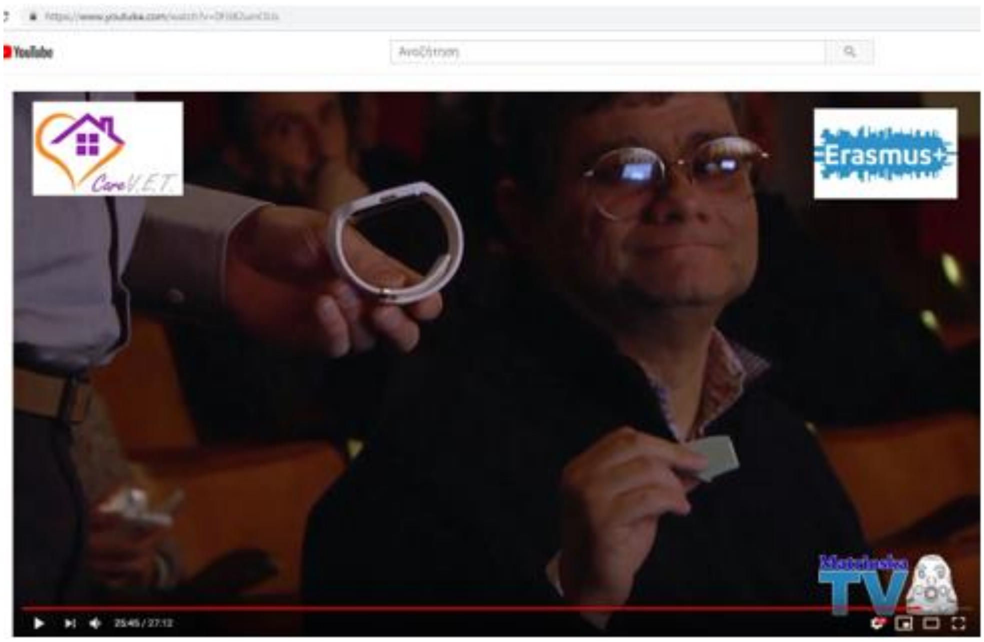


Figure 5: Presenting CareVET toolkit in Rome's conference

Figure 6: Presenting CareVET toolkit in Rome's conference

The toolkit was used during the trainings with trainees to have hands on experiences.

Figure 7: Maurizio Panzironi teaches about the CAreVET toolkit in Italy

Figure 8:Maurizio Panzironi teaches about the CAreVET toolkit in Italy

2.5 Addition of more Sensors

The addition of more sensors is possible in a few steps. Several tests took place with a custom-made Printed circuit board (PCB) supporting a number of different sensors.

Figure 9: addition of more sensors

A custom wearable device was also developed to replace the Android Smartwatch offering more flexibility to the CareVET toolkit and longer battery life.

2.6 Operation

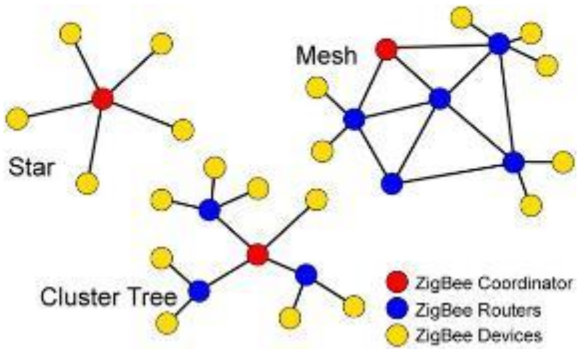
Since you have setup the areas in a living space and the rules that lead to alerts (see Output 3), you can operate the toolkit and receive alerts when things are taking place.

Figure 10: Alerts

3 Technologies

3.1 Zigbee

Zigbee is a secure, low power wireless protocol designed for Internet of Things and home automation solutions. It is based on the IEEE 802.15.4 specification and also implements 128-bit AES encryption for data transfer. Due to its aim for low power consumption – and thus long battery life – data rate is limited to 250 kbit/sec and the transmission distance can be from 10 to 100 meters line-of-sight depending on the configuration of the device.



The protocol makes use of mesh and star topologies and the connected devices can be coordinators, routers or end-devices. Coordinators and routers can form mesh topologies and can either be always on or turn on and off in sync, with coordinators keeping the time. End devices are usually sensors that periodically wake up to report measurements – these devices connect to routers or coordinators forming a star topology.

Zigbee is ment to be used for transferring small amount of data at low rate, over short disances such as home networks, aiming to be energy consumption friendly. It fits perfectly a wireless sensor network's needs, as there is no need for high data rates and the volume of data being transferred is usually low indicating simple things like if a door is open or reporting a numerical measurement such as temperature. Wireless sensors need to be energy efficient as they usually operate on battery power. With a Zigbee wireless network low energy consumption can be achieved leading to prolonged battery life, often lasting for months. Zigbee is not a widely used network protocol but it is an open one and it has been around for a long time, as well as utilized by some big names such as Philips.

Choosing to include a Zigbee wireless network with the toolkit one can talk about the needs of data transferring for an Internet of Things solution as well as demonstrate the way data flows for its different topologies and compare with the rest of the wireless technologies. Another unique trait of Zigbee that makes its use interesting is the way devices behave when they switch from a low powered state to a normal one in order to send data: it is a fast process introducing minimal delays and thus the devices spend less time in normal mode, allowing for even lower energy consumption.

3.2 Wi-Fi

Wi-Fi is a wireless protocol based on the IEEE 802.11 specification. It was introduced 20 years ago and has been through many revisions that allowed for better range and faster data rates as the protocol matured. Nowadays it can be found everywhere, and because of this it is the main wireless protocol used for Internet of Things. Although mesh networking support in starting to appear for Wi-Fi networks, it is not widely available yet and the vast majority of Wi-Fi networks utilize the star topology. That means that all the devices need to connect directly to a Wi-Fi access point for Internet. Wi-Fi can use WEP, WPA or WPA2 security as well as a certificate-based system following the 802.1X standard. Most residential Wi-Fi infrastructure uses WPA2 with a 256-bit security key, while enterprises usually choose certificate-based security.

Wi-Fi was designed to be fast and data rates could even reach 1Gbit/sec with the latest protocol revisions, but are usually around 100-200 Mbit/sec. The range of Wi-Fi networks is around 40 meters indoors and almost 100 meters outdoors. These numbers vary depending on the obstacles and the frequency used, as Wi-Fi can use both 2.4GHz and 5GHz bands.

It should be obvious that Wi-Fi is very well suited to transferring large amounts of data and is usually used for file transferring and audio and/or video streaming, even in real time. As discussed in the previous section, Internet of Things most often needs not fast data rates as sensors do not send or receive large amounts of data. Wi-Fi is fast and that also has a negative impact on battery. In addition, in contrast to Zigbee devices, Wi-Fi devices' switching between low powered and normal state is slow causing devices to operate in normal mode for longer times taxing their battery even more.

These are questions that the toolkit would like to train one to think about, arriving to the conclusion that Wi-Fi might not be the best solution for Internet of Things solutions. However, Wi-Fi is everywhere and that makes it the easiest solution as one does not need to install anything extra to deploy an Internet of Things network. The toolkit aims to lead one's train of thought to a more precise conclusion that takes in account the advantages and disadvantages of Wi-Fi for Internet of Things networks: availability versus energy consumption - so for devices that do not need to communicate wirelessly that often, Wi-Fi could be the better solution.

3.3 Bluetooth

Bluetooth is based on IEEE 802.15.1 specification and is a wireless protocol that seems to combine characteristics of Zigbee and Wi-Fi. Bluetooth has been around for a while and has gone through 5 main revisions so far (Bluetooth 5.0 is the latest one as of this writing). With revision of Bluetooth 4.0 the Bluetooth Low Energy, also known as BLE, was introduced.

Bluetooth uses the same frequency as Wi-Fi at 2.4GHz for data transfer at rates of 25 Mbits/sec. Note that this speed is faster than Zigbee, though slower than Wi-Fi. Bluetooth range highly depends on the device's implementation and power configuration and is similar to Zigbee: 10 – 100 meters.

BLE is quite similar to Zigbee and was designed for low energy consumption. All devices that are BLE capable are also capable of normal Bluetooth communication. This allows for choosing between two states of operation depending on each device's needs: BLE offers better battery management with low data rates (like Zigbee), while normal Bluetooth is faster though more energy demanding.

BLE can allow for extra low energy consumption modes where a device can only transmit a very small amount of data while allowing for detecting requests for normal Bluetooth connection and switch to that when needed. This very low power state with BLE is the perfect fit for advertising a device's ID. This device can serve as an indoors location node: a properly configured mobile node can pick up the advertised ID and look it up at a table providing the location of reference nodes, thus finding out its own location on an indoors location map. That is how BLE is demonstrated in the toolkit, showing off the potential of such technologies.

3.4 Raspberry Pi

Rasperry Pi, also known simply as rPi is an embedded platform that can substitute a fully functioning PC. Its main operating system is Linux, although there are versions of Windows and other operating systems that support it.

Pictured is the Raspberry Pi Zero W, which is the minimal version of the platform in ways of physical external connections, but – as denoted by the W in the name, it has wireless connectivity to make up for that. It supports Bluetooth and Wi-Fi networks and the toolkit provides Zigbee connectivity through a USB dongle.

The Raspberry Pi has the role of the coordinator in the Zigbee network it creates as well as the role of the internet gateway. It can also act as an indoors location reference node, utilizing its Bluetooth with BLE capabilities.

As part of the toolkit it is setup with the Raspian Linux Operating System and with linux scripts it automatically brings up the zigbee network while connecting to Wi-Fi for internet access. The Zigbee network is managed through python scripts that run upon boot completion.

3.5 Android Smartwatch

A smartwatch is like a miniature of a smartphone, but it usually does not offer cellular connection directly. It can simulate that by communicating to cellular networks through a companion smartphone. Nowadays, smartwatches have Wi-Fi and Bluetooth connectivity. Most of them also support the latest Bluetooth Low Energy version as well. For the toolkit an android smartwatch was used that had both Wi-Fi and BLE connectivity. It was used as the mobile node for a simple indoors location solution.

Android is an operating system based on the Linux operating system. The android operating system provides easy access to the device's functions, such as Wi-Fi and BLE configuration and utilization through API calls. Android also allows for easy battery management with the use of background processes and screen brightness control. Applications for Android OS are programmed in the Java programming language or C++ depending on the app's needs. In the toolkit's case Java was preferred as it has a larger user base for android, and with the most recent updates performance is close to C++ too.

The android smartwatch is part of an indoors solution implemented for the toolkit to highlight several aspects of Internet of Things platforms and emphasize challenges of wireless technologies. The smartwatch is the mobile node for the indoors location system and scans for BLE advertising nodes – the reference nodes. Then using Wi-Fi reports the ID with the strongest signal reception, naively assuming that it is the nearest one, and receives from the server its location. The server has a table with the reference nodes' locations that a user can configure online.

This procedure can be a study case for one to understand how information flows and ways that Internet of Things solutions can use more than one wireless network. In addition, one can experience how errors can result from interference: for example covering the smartwatch with one's hand could lead to reporting the wrong reference node as the closest due to signal absorption by the human body. Building the system to be as simple as possible allows for easier understanding while provoking questions such as why use BLE for reference nodes that move the teaching experience forward.

3.6 Python and Java

Python and Java are two widely used programming languages. Python has risen to be the language of choice for quick and easy development and for this reason is the go to language for the Raspberry Pi platform. Java is powerful while simple enough and is the main language used Android app development.

Python and Java are cross platform and are both extensible through packages. They both enjoy a large growing user base thus support is very good. For the needs of the CareVET toolkit, Python was used in Raspberry Pi to parse and respond to Zigbee network's demands. Java was used for developing the location app for Android smartwatch.

Following a code extract to automatically find the USB Zigbee dongle port for Raspberry Pi:

```
#find netvox usbSerialPort
def findPort(driver):
    results = []

    out = check_output(["bash"," ls -l /sys/class/tty/*/device/driver"])
    out = out.split("\n")

    for p in out:
        if driver in p:
            p = p.split("/")
            for n in p:
                if "ttyUSB" in n:
                    results.extend(["/dev/"+n])
    return(results)
```

The following code extract performs Wi-Fi post to the server with data from Zigbee sensors:

```
import requests
import datetime as dt
import xml.etree.ElementTree as ET
import sys

now = str(dt.datetime.now())
url = ""
sensorIEEE = ""
parentIEEE = "NULL"
sensorType = ""
attribute = []
attributeValue = []

sys.argv.reverse()
sys.argv.pop()
if len(sys.argv) >= 6:
        url = str(sys.argv.pop())
        sensorIEEE = str(sys.argv.pop())
        parentIEEE = str(sys.argv.pop())
        sensorType = str(sys.argv.pop())
        attribute.append(str(sys.argv.pop()))
        attributeValue.append(str(sys.argv.pop()))
else:
        print "!!: Warning not enough/correct args, running with default values!"
        print "!!: __ args should be: postURL IEEE parentIEEE sensorType Attr1 Attr1.Value Attr2 Attr2.Value ... ____"
        print "!!: give NULL as parentIEEE to ignore it"
        exit(-1)

while len(sys.argv):
        attribute.append(str(sys.argv.pop()))
        attributeValue.append(str(sys.argv.pop()))

if parentIEEE=="NULL":
        dataSW = """<SenseWorks>\r\n\t<sensordata IEEE="{0}" type="{1}">\r\n""".format(sensorIEEE,sensorType)
else:
        dataSW = """<SenseWorks>\r\n\t<sensordata IEEE="{0}" parentIEEE="{1}"
type="{2}">\r\n""".format(sensorIEEE,parentIEEE,sensorType)
while len(attribute) and len(attributeValue):
        dataSW += """\t\t<{0} str_Datetime="{1}">{2}</{0}>\r\n""".format(attribute.pop(),now,attributeValue.pop())
dataSW += """\t</sensordata>\r\n</SenseWorks>"""

response = requests.post(url, data=dataSW, headers=headers)
print response.content
```

Also some code extract in Java from the smartwatch Android App. The code extract that follows is the construct that holds BLE reference nodes that were in range while scanning.

```
import java.text.SimpleDateFormat;
import java.util.Calendar;

public class BleDevices {
    public String name;
    public String mac;
    public int rssi;

    private String time;
    private Calendar cal = Calendar.getInstance();
    private SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

    public String getTime()
    {
        return time;
    }

    public String getRssi()
    {
        return String.valueOf(rssi);
    }

    public BleDevices() {
        name="unset";
        rssi=0;
        time = "";
    }

    public BleDevices(String devName, String mac, int Rssi){
        this.name=devName;
        if(this.name == null){
            this.name = "";
        }
        this.mac = mac;
        this.rssi=Rssi;
        time = df.format(cal.getTime());
    }
}
```

And the following code extract manages the BLE scanning for devices nearby and storing them in the construct described above.

```
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothManager;
import android.bluetooth.le.BluetoothLeScanner;
import android.bluetooth.le.ScanFilter;
import android.bluetooth.le.ScanSettings;
import android.content.Context;
import java.util.List;
import java.util.Queue;
import java.util.concurrent.ConcurrentLinkedQueue;

public class BleManage {
   private Context mContext;

   private BluetoothAdapter mBluetoothAdapter;
   private static final long SCAN_PERIOD = 10000;
   private BluetoothLeScanner mLEScanner;
   private ScanSettings settings;
   private List<ScanFilter> filters;

   public static Queue<BleDevices> data = new ConcurrentLinkedQueue<BleDevices>();

   BleManage(Context activityContext)
   {
      //init stuff here
      mContext = activityContext;
      mBluetoothAdapter =
((BluetoothManager)mContext.getSystemService(Context.BLUETOOTH_SERVICE)).getAdapter();

   }

   public void scanLeDevice(final boolean enable) {
      if (enable) {
         mBluetoothAdapter.startLeScan(mLeScanCallback);
      } else {
         mBluetoothAdapter.stopLeScan(mLeScanCallback);
      }
   }

   private BluetoothAdapter.LeScanCallback mLeScanCallback =
         new BluetoothAdapter.LeScanCallback() {
            @Override
            public void onLeScan(final BluetoothDevice device, int rssi,
                         byte[] scanRecord) {
               final int mRssi = rssi;
               ((Activity)mContext).runOnUiThread(new Runnable() {
                  @Override
                  public void run() {
                     //add device to queue
                     BleDevices t = new BleDevices(device.getName(), device.getAddress(), mRssi);
                     data.offer(t);
                  }
               });
            }
         };
}
```

4 ANNEX I: Flyer for the Toolkit

CAREV.E.T. TOOLKIT

What is it?

The toolkit combining location information from a smart watch and stationary Bluetooth beacons and data from motion and intrusion sensors offers a solution that can deliver adaptive services with regard to the knowledge of users' lifestyle. The toolkit is easy to install, requiring only to place the sensors in the living environment and to provide the local Wi-Fi Password through a web interface or a smart watch Android app. It is indented for training purposes.

Bluetooth Low Energy (BLE) Beacons

Android Smartwatch

Door and Motion Sensors

Raspberry Pi Gateway

Rule Based Monitoring Application

PROJECT CAREVET

CareV.E.T. project is funded by Erasmus+ programme. The title of the project is "Innovative training approach for Supportive Living Operators" and its number is: 2016-1-EL01-KA202-023612.

Erasmus+

European Commission

ΙΔΡΥΜΑ ΚΡΑΤΙΚΩΝ ΥΠΟΤΡΟΦΙΩΝ IKY

Care V.E.T.

The purpose of the development of the CareV.E.T. Toolkit is to help trainees to:

- Understand Internet of Things (IoT).
- Assess whether a sensor is working properly.
- Understand when and why a specific wireless network is preferred.
- Correctly install and utilize sensors.
- Discover what kind of power source each sensor is using and why.

Toolkit as a teaching aid can be helpful to understand:

- How is information transmitted wirelessly?
- Experience the way information flows from sensor to cloud server.
- What are embedded devices?
- How Internet of Things (IoT) combines wireless networks and embedded devices.
- How indoors vs outdoors location works.

The Toolkit comes with an online application that sends alerts when rule parameters are satisfied. This way the Supportive Living Operator is alerted for the daily lifestyle of her beneficiaries.

See more at: www.carevet.eu